23rd International Conference on Renewable Energies and Power Quality (ICREPQ'25)
Tenerife (Spain), 25th to 27th June 2025
ISBN-13: 978-84-09-66511-2

Simulation and design software for photovoltaic self-consumption systems

J. Gomez-Cornejo[1], I. Lopez[1], I. Aranzabal[1] , J.M. Guerrero[1] and G. Casado[1]

[1] Department of Electrical Engineering
E.I.B., University of the Basque country UPV/EHU
Plaza Ingeniero Torres Quevedo, 1, 48013 Bilbao (Spain)
Phone: +0034 601 4190, e-mail: julen.gcb@ehu.eus

Abstract.

This work presents the developed software for designing photovoltaic self-consumption systems using free and open-access and Python based Oemof packages. The program allows users to design their own systems, specifying variables, and obtaining simulation results. In includes de possibility of incorporating storage systems that can optimize performance and energy flows, making the systems more efficient and cost-effective. The software enables to analyse various scenarios, including grid interaction and battery storage, to determine the best configurations for different conditions. The approach aims to help reducing energy dependence on traditional resources, promoting the use of renewable energy, and providing significant economic and environmental benefits to communities. By enabling users to actively participate in the design process, the software fosters greater autonomy and supports the creation of sustainable energy solutions tailored to specific needs. The software has been validated by performing several simulation cases, promising effectiveness and reliability for real-world applications.

Key words. photovoltaic, self-consumption, Python software, system design, renewable energy.

1. Introduction

Renewable energies, like solar, wind, and hydro power, are crucial in combating climate change [1] as they reduce greenhouse gas emissions, decrease reliance on fossil fuels, and promote sustainable development. Among these, solar power is one of the most interesting alternatives [2], particularly for residential areas, cities, and energy communities. This technology not only empowers consumers by allowing them to generate their own electricity but also contributes to a more resilient and eco-friendly energy infrastructure.

Designing the most efficient solution for each case requires a personalized approach to selecting and configuring the resources or devices to be used. This ensures that the specific energy needs and environmental conditions of each location are met, optimizing performance and maximizing the benefits of renewable energy systems. The ability to perform simulations of distinct scenarios further enhances this process by predicting outcomes and refining designs.

Designing tools for PV systems often rely on complex software that typically comes with a cost and can be difficult to customize. These limitations can hinder accessibility and flexibility, making it challenging for users to adapt the tools to their specific needs. To address these issues, the aim of this work has been to develop an open-source software that provides a free designing tool. This tool can be tailored to meet specific requirements, enabling anyone interested to use it without financial barriers. By offering a customizable and cost-free solution, this software democratizes access to efficient PV system designs, fostering innovation and broader adoption of renewable energy technologies. This approach not only empowers individual users but also supports the broader community in advancing sustainable energy solutions.

The article is structured as follows: The first section reviews the most remarkable software tools for designing PV systems, highlighting their features and limitations. Next, the second section presents the designed solution, detailing both the Oemof (Open Energy Modeling Framework) packages utilized and the developed software. Following this, the third section focuses on the validation process, demonstrating the functionality and reliability of the proposed tool. In addition, it discusses the results obtained from the performed simulations and case studies. Finally, the conclusion section summarizes the aspects of the developed solution and outlines potential future study lines to further enhance the software and its applications.

2. Existing software alternatives

In this section, the most remarkable existing software tools for designing PV based systems alternatives are discussed.

To begin with, HOMER [3] (Hybrid Optimization of Multiple Energy Resources) is a powerful software tool for modelling and optimizing hybrid renewable and distributed generation systems. It enables to simulate various energy configurations to find the most cost-effective and reliable solutions. It includes three products:

Homer Grid, Homer Front, and Homer Pro. PV*SOL [4] is another professional software tool for designing and simulating PV systems. It enables to design and optimize solar installations with detailed configuration, shading analysis, and 3D visualization. Additional features include financial forecasting, battery storage system planning, and support for electric vehicles. Continuing with paid alternatives, ETAP (Electrical Transient Analyzer Program) [5] ETAP (Electrical Transient Analyzer Program) is a versatile software used for designing, analysing, and operating electrical power systems. It offers functionalities like power flow analysis, short circuit analysis, and transient stability. ETAP is popular in industries such as utilities, industrial plants, and renewable energy. Finally, PVsyst [6] is a widely used comprehensive software for designing and simulating PV systems. It offers detailed shading analysis, energy yield prediction, and financial modelling.

Overall, these mentioned alternatives come with a significant financial cost, which may limit accessibility. In addition, due to their complexity they usually require specific formal training to utilize their full potential effectively. In this sense, HelioScope [7] can be a more user-friendly web-based alternative for designing and selling solar PV systems (including 3D design, energy yield simulation, and financial calculations).While it is also a paid software, it may have limited functionality for highly customized designs.

On the free software side, several alternatives are available, such as, SAM (System Advisor Model) [8] and PVWatts [9] developed by the National Renewable Energy Laboratory (NREL) from USA or OpenSolar [10], a free software developed by a team of solar industry veterans with the aim to empower the adoption of solar energy globally. Furthermore, PVGIS provides data on solar irradiation and energy performance and is a free online tool developed by the European Union, requiring no installation. These free software alternatives are interesting tools that can aid in the design of PV systems. However, they come with limitations and can be difficult to customize. In this sense, PVlib [11] is a customizable open-source library designed to simulate the performance of photovoltaic (PV) energy systems and available in two versions: PVlib-python and PVILB for Matlab, both of which have grown significantly through contributions from an active community of users. Developed initially at Sandia National Laboratories, PVlib offers a comprehensive set of functions and classes for modelling various aspects of PV systems, including solar position, irradiance, and system output. However, PVlib is specifically tailored for PV system modelling, which may limit its applicability for other types of renewable energy systems.

Considering the limitations of the mentioned alternatives, this work focuses on exploiting the potential of Oemof [12] libraries. These libraries offer a flexible and comprehensive solution for modelling and optimizing energy systems, including but not limited to photovoltaic (PV) systems. Oemof supports the simulation of a variety of power sources, loads, and systems, making it a versatile tool for enhancing the efficiency and accuracy of energy system design. By leveraging Oemof, the program developed in this work addresses key challenges such as variability in solar energy production, consumption and integration.

3. The designed software

The designed software is an open-source, Python-based solution that utilizes Oemof libraries. Oemof, is an open framework for developing energy system models and performing analysis. It consists of a collection of Python libraries, making it an accessible and free application for energy modelling. Its high-level built-in data structures and dynamic typing make it suitable for application development and scripting. In addition, Python is a versatile programming language used for various functions, and its simple and easy-to-learn syntax, along with the absence of a compilation step, makes the edit-test-debug cycle very fast. This makes this solution particularly interesting for non-expert users to understand, use, and customize. Another remarkable feature is that it supports modules and packages, which promotes program modularity and code reuse. During the development of this work, the Demandlib, Feedinlib and Oemof.solph libraries have been particularly important. For this reason, the most relevant aspects of these packages will also be introduced.

The main function of the Demandlib package is to generate load profiles for electricity and heat demand, with this work focusing on the electrical part. The profiles are obtained by scaling BDEW (Bundesverband der Energie- und Wasserwirtschaft) profiles to the desired annual demand, with adjustments made for different types of demand (households, weekend operation buildings or weekday operation buildings). Since the package typically generates annual profiles, parameters have been adapted to develop profiles for any period, from days to years. The function ElecSlp from the BDEW package has also been used, requiring additional parameters such as study dates. The profiles are calculated with data every 15 minutes, converted to kWh, and graphed for the specified period. Additionally, the program accounted for the number of days in the study period and adjusted for leap years. The final demand profiles are scaled based on the number of households or the usable floor area of buildings, ensuring accurate representation of energy consumption patterns.

The Feedinlib module, part of the Oemof group, operates independently and has significant potential for improvement and enhanced functionality. Its primary function is to calculate time series for photovoltaic (PV) and wind energy feed-in. This work focuses exclusively on the PV aspect, using Feedinlib to calculate the potential power output based on provided data. The process is divided into two main parts: obtaining meteorological data to determine solar irradiance and calculating PV generation. Feedinlib provides interfaces for downloading meteorological data, such as open_FRED, which are used to calculate PV production. These interfaces support pvlib and windpowerlib libraries, but only pvlib has been used in this work.

Feedinlib also includes technical parameters for many PV modules, inverters, and wind turbines, which are used in system calculations. After obtaining meteorological data,

the PV generation is calculated for the selected data. Three functions have been created for this process: selecting PV modules and inverters, calculating meteorological data, and calculating system power output. The PV selection function displays different types and selects those closest to the expected parameters, though Feedinlib's limited database may cause calculation errors. The meteorological data function, requires input parameters like study dates and the number of PV modules, generating system variables, meteorological data, and site coordinates using Feedinlib's Photovoltaic function. Finally, the power output function, calculates the generated power, converting output data to kilowatts and graphing power over time for visual representation.

The oemof.solph package is a model generator for modelling and optimizing energy systems, designed to create and solve linear or mixed-integer linear optimization problems. In this case, among the existing paid and free solvers, the recommended one, CBC (Coin-or Branch and Cut), has been utilized. The initial step involves defining an EnergySystem variable as the main container for the energy system model, with a Datetime index to set the time range and increment. The process of building the model includes defining various components, which are interconnected to buses through flows. Buses represents a lossless grid or network, with balanced inflows and outflows.

Components are categorized into basic (solph.network), additional (solph.components), and custom (solph.custom). There are three basic component types: Sink, Source, and Transformers. The Sink component has been configured to define the energy demand within the model, ensuring the connected bus meets this demand, and to detect energy excess, directing surplus energy to the grid for sale with an assigned cost variable representing the sale price. This component has been connected to the main bus where general calculations are performed The Source component has been used to represent photovoltaic systems, defining the generated power and transferring it to the connected bus. This component has been employed to model the power output of solar panels, ensuring the energy flows from the Source to the bus. The power generated by the source component is specified, and this energy is transferred to the connected bus. The Transformer component acted as a node with multiple input and output flows, linking subsystem buses with the main system bus, allowing bidirectional energy flow. For each connection between a subsystem bus and the main bus, two Transformer components have been used: one for the flow from the main bus to the subsystem bus, and another for the flow in the opposite direction.

Among the additional components, the GenericStorage type has been used to model a battery system. The nominal energy rates in kWh have been defined, varying by building. The GenericStorage components have an input and output to allow batteries to supply energy to the bus when needed and store excess energy. The input has been connected to the bus with specified charging power in kW, and the output with discharging power in the same units. Both have been assigned a minimal cost parameter of 0.001, prioritizing battery storage over grid transactions. Despite advancements, battery costs remain high, complicating amortization. The developed tool offers the possibility to perform analyses for decision-making regarding whether to install the battery or not, and if so, its sizing.

Figure 1 illustrates the general diagram of basic system. The left side shows buildings and their components. Although the diagram represents one building, the program supports multiple (n) buildings. Each building has similar components. The bus "Electric-node N" receives energy from photovoltaic generation and the building's battery if available. It supplies energy to meet the building's demand and enables to store the excess energy in the batteries. To connect each building's bus to the main bus ("Electric-node"), “Transformer input" transmits energy from the building to the main bus, and “Transformer output" does the reverse. The right side includes elements independent of the number of buildings, such as the main bus. The program enables to install a "General battery" for the entire system, in addition to individual building batteries.

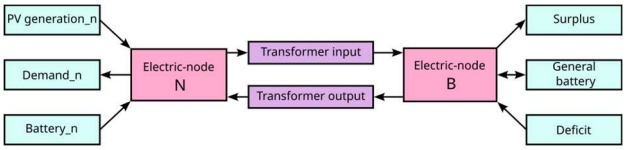


Fig. 1. General diagram of a basic system

The system connects to the grid to buy “Deficit" or sell “Surplus” energy. Excess and deficit represent the system's grid connection. If there is a deficit, energy is purchased from the grid. Similarly, excess energy can be sold to the grid. Both components connect exclusively to the main bus. Prices are obtained from a database file with prices defined by the E·SIOS [13] website.

The user specifies whether to use battery storage and sets parameters like nominal capacity, charge power, and discharge power. Building-specific elements determine demand, quantifying building size. For residential buildings, the parameter is the number of dwellings, which the program asks the user to specify. This value is stored in a variable and multiplied by the demand obtained from the "Demand" function. Power data is calculated using meteorological data from a function that processes the irradiance input provided. Input parameters like start and end dates must also be defined. The output includes system location and meteorological data.

Once the system has been defined, optimization and solution processes begin. The system is first solved using the installed solver (CBC). Both the general case, which solves each element separately, and the meta case, which solves the system as a whole, are addressed, and results are restored. Data is then processed to generate results represented through graphs. These steps are repeated for all defined buses, including the battery systems of buildings and the general battery. The process concludes with the system solution and graphical representations of each element's results. The system has been validated through various tests, examining how results vary with different system parameters.

4. Validation and results

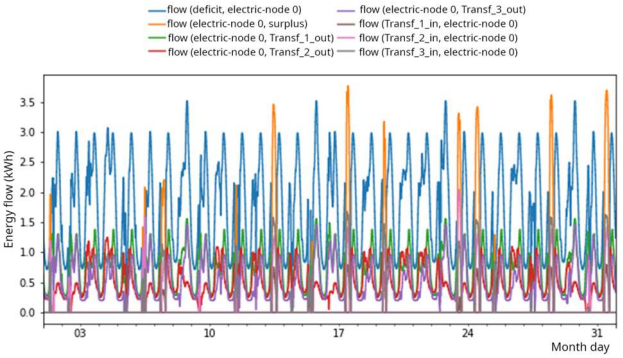
This section focuses on the validation process, demonstrating the accuracy and reliability of the proposed tool. It discusses the results obtained from the performed simulations and case studies. In this part, an analysis of the program's functions has been carried out by examining different possible scenarios and analysing the results obtained from them. It is important to note that although multiple studies can be conducted by varying input parameters and modelled systems, the analysis has been limited to a few specific scenarios, which enable to prove the program's functionality, and the possibilities it offers.

Despite the existence of various study variables, several variables have been defined as constants for both cases. The installation location chosen is the city of Bilbao, with geographical coordinates of latitude 43.263 and longitude -2.925. The orientation of the photovoltaic panels has been set to 180 degrees in all cases.

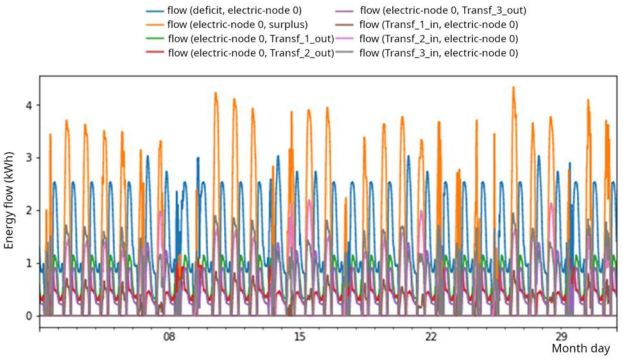
Both cases with and without battery-based electrical storage have been analysed. In the case of batteries, the use of individual batteries per building and a general battery have been examined. In addition, it has to be remarked that although the graphs of the different buses and buildings have been obtained, only the operation of the connection bus will be shown in this case as it represents the most significant result, streamlining the explanation and keeping the article concise.

In the first test, a scenario of three buildings without a battery storage system has been defined: building 1 is residential, building 2 is a workplace with a weekday schedule, and building 3 is a workplace with a predominant weekend schedule. For the residential building, one with 8 apartments and 10 photovoltaic modules has been selected. For the office buildings, 400 square meters of usable space has been taken as the value, and 15 modules have been installed.

For the sake of simplicity, among the different test performed, only the results for the two most significant months of the year are presented: January for winter and August for summer. In this way, this analysis shows the behaviour in the months with more different meteorological conditions and generation levels. The results for this simulation are shown in Fig. 2. As can be observed the system deficit curve (blue), representing energy purchases from the grid, has predominated over the excess curve (orange), which represents energy sales to the grid. In August, Fig.2 (b), the opposite occurs due to the generation profile varying by month. Despite having more excess than deficit in August, both curves constantly vary, frequently switching between excess and deficit. This has been studied in subsequent cases, considering the introduction of photovoltaic generation (green) and energy exchange with the connection bus. The orange curve represents the flow from the residential building bus to the connection bus, while the red curve represents the opposite direction. Demand peaks more in January, and generation is higher in August. The flow to the connection bus has been limited in the first case due to insufficient energy, while the energy requested from the general system has been higher. In August, the flow to the general connection has been higher due to increased generated energy, but the requested energy curve has also been significant, though smaller than in the first case. Similar situations have been observed for the other two buildings, with higher generation and greater energy contribution to the system.



(a)



(b)

Fig. 2. Connection bus flows for the cases without a battery system: January (a) and August (b)

For the cases of systems with battery storage the same location and the same panels and orientation have configured. The system chosen for this case study has been slightly simplified compared to the previous one, consisting of two buildings. The first is a commercial building with a weekday schedule, covering 200 square meters and equipped with 18 photovoltaic modules. The second is a residential building with four apartments and 15 modules.

The first scenario analysed includes a battery system (capacities of 6.75 kWh nominal energy and 3.36 kW charging and discharging power) installed in the buildings. The results for this simulation are shown in Fig. 3 (in order to preserve the length of the manuscript only the results for August are presented.) As it can be observed, the excess curve has predominated, indicating that more energy has been generated than consumed, and the battery capacity has been insufficient. Although the deficit is small, it exists and could have been eliminated with higher capacity batteries. The high final excess is due to the inadequate battery capacity for the given conditions. The energy flows from the buildings to the connection bus have been substantial, and with small battery systems and high generation, the only option for the excess energy has been to sell it to the grid. Despite the generation being much higher than the demand, there have been deficit periods,

also due to the incorrect battery size selection. With larger capacity batteries, the need to purchase from the grid would have been avoided.

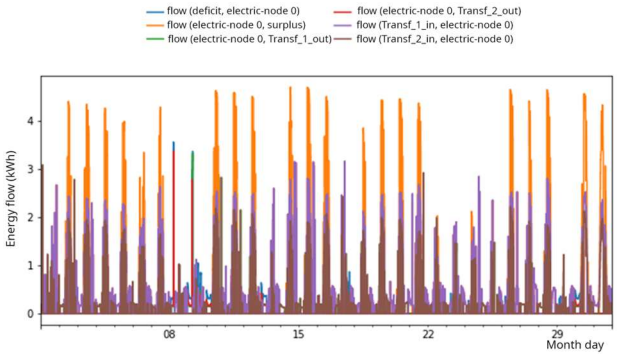


Fig. 3. Connection bus flows for the case of batteries installed in the buildings (August)

The next case studied has been the installation of a general battery system replacing the two batteries located in each building with a large general battery (nominal energy of 40.5 kWh and charging and discharging power of 19.2 kW). In this case, Fig. 4 presents the flows and storage of the general system. The storage curve (orange) initially did not fully charge, suggesting it might be too large, but in the final days, the battery capacity is utilized. This indicates that the selected parameters are correct and provide the desired benefits for the installation's operation. Additionally, the deficit has been eliminated, and the excess has been significantly reduced, resulting in a more efficient system that no longer requires grid purchases. The remaining excess is due to higher generation than needed.

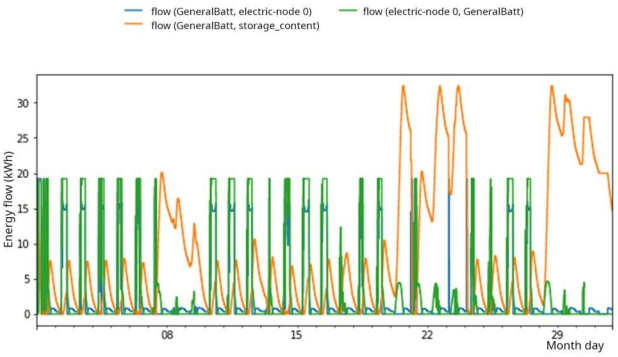


Fig. 4. Energy flow for the case of a general battery system (August)

The last scenario presented in this work represents a more complex system since it includes, a general battery system (nominal energy of 27 kWh and charging and discharging power of 13.44 kW) and three buildings with and without an individual battery system. In this way, the first is a residential building with 6 apartments and 12 photovoltaic modules, equipped with a storage system (13.5 kWh nominal energy and charging and discharging power of 6.72 kW). The second is another residential building with eight apartments and eight modules. The third is a commercial building with a predominant weekend schedule, covering 300 square meters and equipped with 18 modules. The period analysed, differing from previous cases, spans from April 1 to mid-May. The graph of the flows and storage of these batteries is shown in Fig. 5. This graph shows that the battery has been operating and able to store energy during periods of excess. At certain times, the battery had low charge, indicating low generation on some days, likely due to the variable weather conditions in April and May, which resulted in insufficient sunlight to meet demand and create an excess. Energy purchases from the grid were minimal, totalling 24.32 kWh. However, the deficit was quite large at 1924.67 kWh, attributed to poor weather conditions for generation or insufficient photovoltaic systems for the buildings' requirements. The battery flows indicate that they were able to operate due to the excess at certain times, reducing the need for grid purchases, which would have been higher without storage.

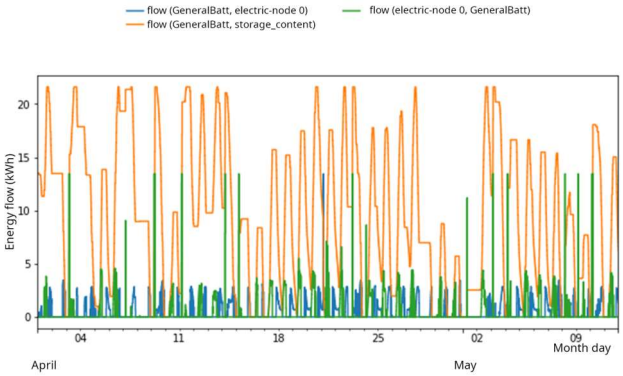


Fig. 5. Energy flow for the case of a mixed battery system

The results obtained from the presented cases demonstrate the functionality and potential of the developed software tool.

4. Conclusions

In this work, a developed software that enables to design and simulate self-consumption systems has been introduced. Developed in Python utilizing Oemof packages, the created program represents an analysis tool providing several functionalities such as integrating self-consuming systems. In addition, this program offers a cost-effective solution as it is free software, eliminating the need for expensive software licenses.

The design process has been extensive, highlighting the importance of accurate demand calculation. Real electrical system data can be more beneficial than theoretical data for final projects, emphasizing the need for precise and reliable input data. The performed analyses indicate that the introduction of storage systems is highly beneficial for these systems, as it helps balance renewable generation in the electrical grid, which is one of the primary objectives of such systems. In this way, the program has proven useful due the obtained successful results.

However, the program has certain limitations, such as a restricted period for solar irradiation calculation and the absence of a larger number of nodes or inverters. Additionally, some Oemof packages still require improvements or updates to facilitate a more comprehensive analysis. Therefore, the future line of research would be to improve the described limitations and

to consider the inclusion of different generation alternatives such as geothermal energy or mini and micro wind energy.

Acknowledgement

This work is financially supported by the Basque Government under the Grant IT1647-22 (ELEKTRIKER research group) and by the Grant TED2021-129930A-I00 funded by MICIU/AEI/10.13039/501100011033 and by the "European Union NextGenerationEU/PRTR". Funded by the European Union also. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the European Climate, Infrastructure and Environment Executive Agency (CINEA). Neither the European Union nor the granting authority can be held responsible for them.

References

[1] Duggan Jr, J. E. (2025). Recent Progress in Sustainable Energy Systems Development: Investment, Operations, and Decarbonization. Current Sustainable/Renewable Energy Reports, 12(1).
[2] Maka, A. O. M., & Alabid, J. M. (2022). Solar energy technology and its roles in sustainable development. Clean Energy, 6(3), 476-483.
[3] HOMER (Hybrid Optimization of Multiple Energy Resources). Available at: https://www.homerenergy.com/
[4] PV*SOL. Available at: https://www.valentin-software.com/en/products/photovoltaics/1/pvsol
[5] ETAP. Available at: https://etap.com./
[6] PVsyst. Available: https://www.pvsyst.com/
[7] HelioScope. Available at: https://www.helioscope.com/
[8] SAM (System Advisor Model). Available at: https://sam.nrel.gov/
[9] PVWatts. Available at: https://pvwatts.nrel.gov/
[10] OpenSolar. Available at: https://www.opensolar.com/
[11] PVlib. Available at: https://pvlib-python.readthedocs.io/en/stable/
[12] Oemof. Available at: https://oemof.org/
[13] E·SIOS. Available at: https://www.esios.ree.es